F-039

マルチタスク特徴抽出アルゴリズムを用いたコスト考慮型SVMに関する検討

A Study on cost-sensitive SVM using multi-task feature extraction framework

本郷 辰哉 † 杉浦 徹 † 烏山 昌幸 †† 竹内 一郎 †
Tatsuya Hongo Toru Sugiura Masayuki Karasuyama Ichiro Takeuchi

1 はじめに

Support Vector Machine(SVM) や最近傍法などの分類アルゴリズムにおいて, 通常は各クラスに対する誤分類のコストを等価として学習アルゴリズムが構築される. しかし, 現実問題では, 誤分類のコストが等価でない問題が存在する. 例えば, がんの診断の場合, 健常者をがんと誤分類するコストとがん患者を健康と誤分類するコストは異なると考えられる. 各クラスの誤分類コストが異なることを考慮した学習法はコスト考慮型学習と呼ばれている [1, 2, 3]. 分類器を学習する際には利用時の誤分類コストが不確定であることが多い. そのような場合, 様々なコスト比に対するコスト考慮型分類器を学習する必要があり, 同一データに対してコスト比の異なった複数の問題を解くことになる. 学習データが共通でコスト比のみ異なる問題は何らかの関連性を有すると考えられ, 関連性を取り入れて学習することにより性能向上が期待できる.

関連のある複数の問題を解くとき, それらの問題を個別で解くより, 問題間の関連性や共通性を利用する方がよい場合がある. 複数のタスクの関連性を利用する学習パラダイムをマルチタスク学習といい, 近年, 盛んに研究されている [4, 5, 6].

本稿では, マルチタスク学習のアルゴリズムとしてマルチタスク特徴抽出 [7] をコスト考慮型 SVM の学習に適用することにより, 従来法よりも性能を向上させることができるか実験的に検証する.

以下では, n 次元縦ベクトルを $\boldsymbol{v} \in \mathbb{R}^n$ と表し, $n \times m$ 行列を $M \in \mathbb{R}^{n \times m}$ と表記する. また, $\mathbb{N}_n$ は 1 から n までの自然数の集合 $\{1, 2, \ldots, n\}$ を表すものとする.

2 コスト考慮型SVM

サンプルサイズを n, データの次元を p とする 2 クラス分類問題を考える. 学習データを $\{(\boldsymbol{x}_i, y_i)\}_{i \in \mathbb{N}_n}$ と表記する. ここで, $\boldsymbol{x}_i \in \mathbb{R}^p$ は p 次元の入力ベクトル, $y_i \in \{-1, +1\}$ はクラスラベルである. 本稿では, 分類器の識別関数を線形関数 $f(\boldsymbol{x}) = \boldsymbol{w}^\top \boldsymbol{x} + b$ に限って議論する.

本研究では, 具体的なコスト考慮型分類器として, コスト考慮型 SVM[2] を用いる. 分類器がクラスが y_i であるデータを $-y_i$ と誤分類するコストを $c(y_i)$ とすると, コスト考慮型 SVM は,

$$\min_{\boldsymbol{w}, b} \frac{1}{2nC}\|\boldsymbol{w}\|^2 + \frac{1}{n}\sum_{i \in \mathbb{N}_n} c(y_i)[1 - y_i f(\boldsymbol{x}_i)]_+ \qquad (1)$$

と定式化される. ここで, $[z]_+ = \max\{z, 0\}$ であり, C は正則化項 (目的関数の第 1 項) と損失項 (第 2 項) のトレードオフを制御する正則化パラメータである.

3 マルチタスク学習

マルチタスク学習の目的は, 関連のある複数のタスクが存在するとき, それらのタスクが共有する構造を学習して性能を向上させることである. 一方, 一般的な学習 (以下では個別学習と呼ぶ) では, 複数のタスクが持つ関連性は無視してそれぞれ個別に学習する. 本稿ではマルチタスク学習アルゴリズムのうち, マルチタスク特徴抽出 [7] を用いる. 同一データに対してコスト比の異なる複数の SVM を学習する問題では, 共通の特徴を利用することで性能向上が期待できる. 本節では, [7] のアプローチを原論文にしたがって簡単に説明する.

3.1 マルチタスク特徴抽出

マルチタスク特徴抽出では各タスクに共有される特徴を学習する. ここで, 特徴とは, 入力変数を組み合わせて新たに作られる変数を指し, 本稿では入力変数の線形変換によって得られる特徴のみを考察する. 複数のタスクで共通した特徴を選択することにより, タスク間の関連性を生かすことが可能となる.

同時に学習するタスク数を T 個とし, あるタスク $t \in \mathbb{N}_T$ の分類識別関数を f_t とする. 表記を簡略にするため, 各タスク $t \in \mathbb{N}_T$ はサイズ n の学習サンプル $\{(\boldsymbol{x}_{ti}, y_{ti}) \in \mathbb{R}^p \times \{-1, +1\}\}_{i \in \mathbb{N}_n}$ を持つとする. ここでまずは複数の

†名古屋工業大学, Nagoya Institute of Technology
††東京工業大学, Tokyo Institute of Technology

タスクで共有する特徴数が入力ベクトルの次元数と等しく p であるとする (後ほど特徴数を減らす枠組が導入される). p 次元入力ベクトルから p 個のそれぞれの特徴への変換を $h_j : \mathbb{R}^p \to \mathbb{R}$ とすると, タスク $t \in \mathbb{N}_T$ の分類器は,

$$f_t(\boldsymbol{x}) = \sum_{j \in \mathbb{N}_p} a_{jt} h_j(\boldsymbol{x}) = \boldsymbol{a}_t^\top \boldsymbol{h}(\boldsymbol{x})$$

と表される. ここで, $\boldsymbol{a}_t = [a_{1t} \ \ldots \ a_{pt}]^\top \in \mathbb{R}^p$, $\boldsymbol{h}(\boldsymbol{x}) = [h_1(\boldsymbol{x}) \ \ldots \ h_p(\boldsymbol{x})]^\top \in \mathbb{R}^p$ である. 線形変換による特徴のみを考えると,

$$h_j(\boldsymbol{x}) = \boldsymbol{u}_j^\top \boldsymbol{x}, \ j \in \mathbb{N}_p$$

と表される. ここで, $\boldsymbol{u}_j \in \mathbb{R}^p$ は特徴 j を定める線形変換係数ベクトルで, 互いに直交するものとする. これらの線形変換ベクトルを各列に並べた行列を $U \in \mathbb{R}^{p \times p}$ とすると, タスク $t \in \mathbb{N}_T$ の分類識別関数は

$$f_t(\boldsymbol{x}) = \sum_{j \in \mathbb{N}_p} a_{jt} (\boldsymbol{u}_j^\top \boldsymbol{x}) = \boldsymbol{a}_t^\top U^\top \boldsymbol{x} \tag{2}$$

と表される.

全タスクに共有する特徴を抽出するため, p 個の特徴のうち, いくつかを選択, 削除する状況を考える. T 個のタスクの p 次元特徴ベクトルの係数 $\{\boldsymbol{a}_t\}_{t \in \mathbb{N}_T}$ を各列に並べた行列 $A \in \mathbb{R}^{p \times T}$ で表すことにする. このとき, $j \in \mathbb{N}_p$ 番目の特徴を削除することは, 行列 A の第 j 行をすべて 0 とすることに相当する. 文献 [7] では, 行列 A の混合ノルム正則化を導入することにより, 全タスクに共通して有用な特徴を抽出する枠組を提供している. この枠組は以下のような最適化問題として定式化される:

$$\min_{A,U} \sum_{t \in \mathbb{N}_T} \sum_{i \in \mathbb{N}_n} L_t(y_{ti}, \boldsymbol{a}_t^\top (U^\top \boldsymbol{x}_{ti})) + \gamma \|A\|_{2,1}^2. \tag{3}$$

ここで L_t はタスク $t \in \mathbb{N}_T$ の損失関数で, 本稿の対象であるコスト考慮型 SVM の場合, 重み付きヒンジ損失関数となる. 第 2 項は正則化項で A の $\ell_{2,1}$ ノルムを用いている. $\ell_{2,1}$ ノルムは, まず, 各行の ℓ_2 ノルムをとり, さらに, それらの ℓ_1 ノルムをとった値を表す. 具体的には, A の j 行を $\boldsymbol{a}^j$ と表すと

$$\|A\|_{2,1} := \sum_{j \in \mathbb{N}_p} \|\boldsymbol{a}^j\|_2 \tag{4}$$

と表される. なお, γ は正則化パラメータで, (3) を解いて得られる特徴数は γ に対して非増加関数となっている.

(3) を直接解くことは難しいため, 最適解が等しくなるような制約付き凸計画問題を解く. T 個のベクトル $\{\boldsymbol{w}_t\}_{t \in \mathbb{N}_T}$ を導入し, これらを各列に並べた行列を $W \in \mathbb{R}^{p \times T}$ とする. また, $p \times p$ の半正定値行列 $D \succeq 0$ を新たに導入する. 詳細は割愛するが, 問題 (3) の最適解は問題

$$\begin{aligned} \min_{W,D} \quad & \sum_{t \in \mathbb{N}_T} \sum_{i \in \mathbb{N}_n} L_t(y_{ti}, \boldsymbol{w}_t^\top \boldsymbol{x}_{ti}) + \gamma \sum_{t \in \mathbb{N}_T} \boldsymbol{w}_t^\top D^\top \boldsymbol{w}_t \\ \text{s.t.} \quad & D \succeq 0, \mathrm{tr}(D) \leq 1, \mathrm{range}(W) \subseteq \mathrm{range}(D) \end{aligned} \tag{5}$$

の最適解と一致することが証明される. ここで, $D \succeq 0$ は D が半正定値行列である条件を表し, $\mathrm{tr}(\cdot)$ は行列のトレースを, $\mathrm{range}(\cdot)$ は行列のレンジを表すオペレータである.

(3) は行列 A と U に関する最適化問題であり, (5) は は行列 W と D に関する最適化問題となっている. 最適解における両者の関係は

$$W = UA, \tag{6a}$$

$$D = U \mathrm{diag}\left(\frac{\|a^j\|_2}{\|A\|_{2,1}} \right)_{j \in \mathbb{N}_p} U^\top \tag{6b}$$

となることが証明される. ここで, $\mathrm{diag}(\cdot)$ は, 各要素を対角成分に持つ対角行列である. (6) の関係から, (3) の最適解 (A, U) は (5) の最適解 (W, D) から得ることができる. また, W は入力ベクトル $\boldsymbol{x}$ のパラメータで, D のランクは学習された特徴数と一致することがわかる.

最適化問題 (5) を解くためのアルゴリズムを以下にまとめる. このアルゴリズムでは W と D の一方を固定して他方の最適化を繰り返すものとなっている. 以下では D を固定して W を更新するステップを W ステップと呼び, その逆を D ステップと呼ぶ. まず, W ステップでは D を固定し以下の最適化問題を解く:

$$\min_{W} \sum_{t \in \mathbb{N}_T} \sum_{i \in \mathbb{N}_n} L_t(y_{ti}, \boldsymbol{w}_t^\top \boldsymbol{x}_{ti}) + \gamma \sum_{t \in \mathbb{N}_T} \boldsymbol{w}_t^\top D^{-1} \boldsymbol{w}_t. \tag{7}$$

W ステップでは他のタスクの影響を受けないため, 各タスクを個別に最適化することができる. D ステップでは, (5) の目的関数の第 1 項に D が含まれていないため, 最適解は以下のように解析的に求められる:

$$D_\varepsilon(W) = \frac{(WW^\top + \varepsilon I_d)^{\frac{1}{2}}}{\mathrm{trace}(WW^\top + \varepsilon I_d)^{\frac{1}{2}}}$$

ここで, 定数 $\varepsilon > 0$ は解を安定させるために加えているが詳細は [7] を参照されたい.

4 特徴を共有したコスト考慮型SVMの同時学習

第 2 節で紹介したコスト考慮型 SVM と第 3 節で紹介したマルチタスク特徴抽出アルゴリズムを統合したものが本研究の提案法となる．誤分類のコスト比のみ異なる複数のコスト考慮型 SVM は同じ学習データに対するタスクとなるのでタスク間で関連のある特徴を学習し利用することで性能向上が期待できる．マルチタスク特徴抽出を用いたコスト考慮型 SVM の解くべき問題は (1) と (3) を組み合わせることで以下のように定式化される:

$$\min_{A,U}\{\gamma\|A\|_{2,1}^2+\sum_{t\in\mathbb{N}_T}\sum_{i\in\mathbb{N}_n}c_t(y_i)[1-y_i\boldsymbol{a_t}^\top(U^\top\boldsymbol{x_i})]_+\}. \quad (8)$$

ここで $c_t(y_i)$ はタスク t においてクラス y_i を誤分類した時のコストである．

5 計算機実験

本節では数値実験により本手法の有効性を検証する。複数の誤分類のコスト比のみ異なるコスト考慮型 SVM をタスクとする場合, それらのタスクには何らかの関連があると予想される．実データを用いた実験によりそのようなタスクに対してマルチタスク特徴抽出を用いることでコスト考慮型 SVM の性能が向上することを確認する．用いた実データは UCI Machine Learning Repository から取得した．使用したデータを表 1 に示す．タスクをクラス別の誤分類のコスト比 $(c_{+1}:c_{-1})=(0.9:0.1)$ から $(c_{+1}:c_{-1})=(0.1:0.9)$ までの 0.1 刻みの 9 つとし, 個別学習とマルチタスク特徴抽出で得た分類器の性能を比較する．データはそれぞれ学習, 評価, テストの 3 つにほぼ等しい割合でランダムに 3 分割し, 50 回の試行における平均と標準偏差 (括弧内) を表 2 に示す．また対応する ROC グラフとその AUC 値を図 1 に示す．コスト考慮型 SVM の性能を計るための評価関数として以下の関数を用いた:

表 1: 使用した実データ

データ名	データサイズ	次元数
Bre.Can.Dia.	569	30
Bre.Can.Pro.	194	33
Parkinson	195	20
Ionosphere	351	33

$$L(f)=c_1\sum_{i\in I_+}I(f(\boldsymbol{x}_i)\le 0)+c_{-1}\sum_{i\in I_-}I(f(\boldsymbol{x}_i)>0).$$

ここで, I_+ は $y=1$ であるデータのインデックス集合であり, I_- は $y=-1$ に対応する．関数 I は引数が真の時 1 を返し, 偽の時 0 を返す関数である．

6 結果と考察

表 2 より今回用いた全てのデータセットに対して個別に学習するよりマルチタスク特徴抽出を用いた方がタスク合計での平均評価値が小さくできていることがわかる．また全データセットに対しマルチタスク特徴抽出の方が標準偏差を小さくすることができている．以上よりコスト考慮型 SVM の学習においてマルチタスク特徴抽出が有効であることが分かる．

参考文献

[1] Shunichi Amari and W Wu. Improving support vector machine classifiers by modifying kernel functions. *Neural Networks*, 12(6):783–789, 1999.

[2] Yi Lin, Yoonkyung Lee, and Grace Wahba. Support vector machines for classification in nonstandard situations. *Machine Learning*, 46:191–202, 2002.

[3] G Wu and E Chang. Adaptive feature-space conformal transformation for imbalanced data learning. In *Proceedings of the 14th International Conference on Machine Learning*, pages 816–823, 2003.

[4] B. A. Turlach, W. N. Venables, and S. J. Wright. Simultaneous variable selection. *Technometrics*, 47:349–363, 2005.

[5] G. Obozinski, B. Taskar, and M. Jordan. Multi-task feature selection. Technical report, Department of Statistics, University of California, Berkeley, 2006.

[6] G. Obozinski, B. Taskar, and M. Jordan. Joint covariate selection and joint sbspace selection for multiple classification problems. *Statistics and Computing*, 20(2):231–252, 2010.

[7] A. Argyriou, T. Evgeniou, and M. Pontil. Convex multitask feature learning. *Machine Learning*, 73(3):243–272, 2008.

表 2: マルチタスク特徴抽出 (同時) と個別学習法 (個別) での各データに対する評価値の平均と標準偏差 (括弧内). 評価値の平均は値が小さいほど性能が良いことを表わす. 結果が優れている方を太字で表わす.

		コスト比 $(c_{+1}:c_{-1})$									
データ名	手法	0.9:0.1	0.8:0.2	0.7:0.3	0.6:0.4	0.5:0.5	0.4:0.6	0.3:0.7	0.2:0.8	0.1:0.9	合計
Bre.Can.Pro	同時	**5.284**	**8.260**	**9.046**	**8.348**	**7.510**	**6.316**	**4.716**	**3.112**	**1.528**	**54.165**
		(1.155)	(1.728)	(2.082)	(1.951)	(1.666)	(1.479)	(1.253)	(0.958)	(0.405)	(12.677)
	個別	5.770	8.708	9.228	8.744	7.910	6.452	4.950	3.404	1.742	56.908
		(1.939)	(1.751)	(1.712)	(1.741)	(1.519)	(1.710)	(1.598)	(1.471)	(1.021)	(14.462)
Bre.Can.Dia	同時	**2.590**	**2.760**	**2.844**	**2.840**	**2.670**	**2.548**	**2.338**	**1.848**	**1.184**	**21.622**
		(0.982)	(1.169)	(1.140)	(1.101)	(1.057)	(0.996)	(0.835)	(0.700)	(0.500)	(8.480)
	個別	2.794	3.132	2.958	3.076	3.050	2.688	2.346	2.044	1.360	23.448
		(0.996)	(1.346)	(1.315)	(1.197)	(1.069)	(0.956)	(0.663)	(0.714)	(0.622)	(8.878)
Parkinson	同時	**1.572**	2.616	3.368	**4.028**	**4.770**	**5.572**	**5.856**	5.160	**3.084**	**36.026**
		(0.607)	(1.020)	(1.139)	(1.014)	(1.209)	(1.131)	(1.318)	(1.456)	(1.099)	(9.993)
	個別	1.594	**2.532**	**3.314**	4.292	5.100	6.056	6.184	**5.148**	3.218	37.438
		(0.935)	(1.037)	(1.134)	(1.098)	(1.145)	(1.485)	(1.644)	(1.666)	(1.688)	(11.832)
Ionosphere	同時	**8.746**	**9.704**	**9.312**	**8.588**	**7.720**	**6.600**	**5.392**	**4.440**	2.918	**63.420**
		(2.506)	(2.363)	(2.074)	(1.867)	(1.628)	(1.233)	(1.084)	(0.997)	(0.755)	(14.507)
	個別	9.216	10.364	9.894	9.312	8.420	7.108	5.904	4.680	**2.774**	67.672
		(2.367)	(2.443)	(2.184)	(1.758)	(1.550)	(1.553)	(1.500)	(1.046)	(0.729)	(15.130)

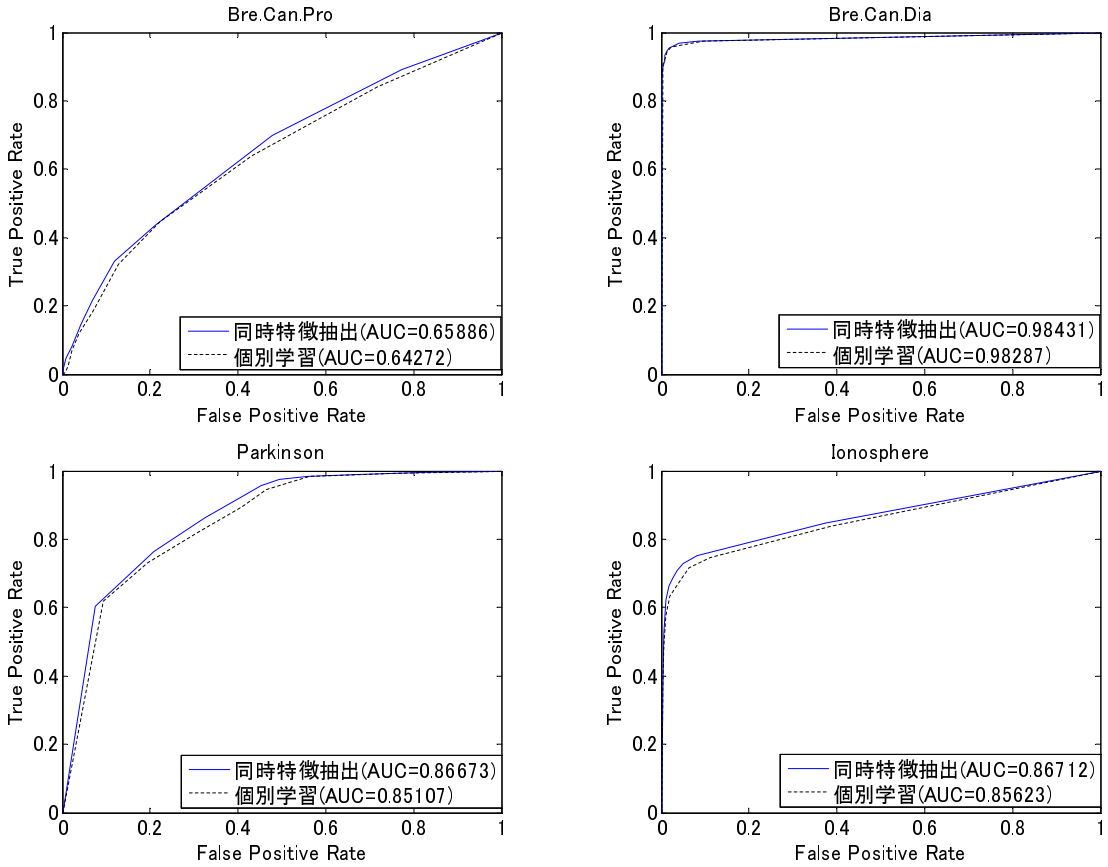

図 1: 両手法の各データに対する ROC グラフと AUC. 横軸は正例を負例と誤分類した割合, 縦軸は正例を正しく分類した割合. ROC グラフは左上にある方が性能が良いコスト考慮型分類器である. AUC は ROC グラフの下の面積である.